

Fig. 2  
Server-side

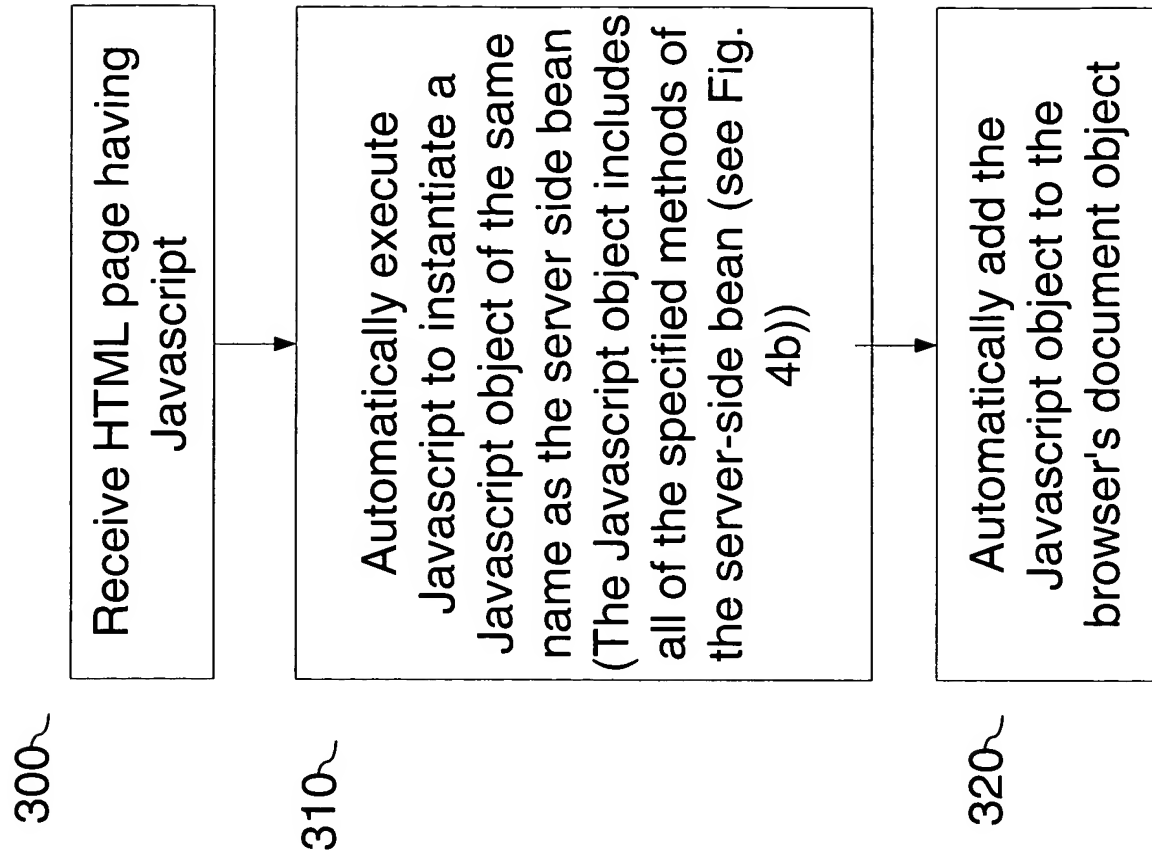


Fig. 3  
Client-side

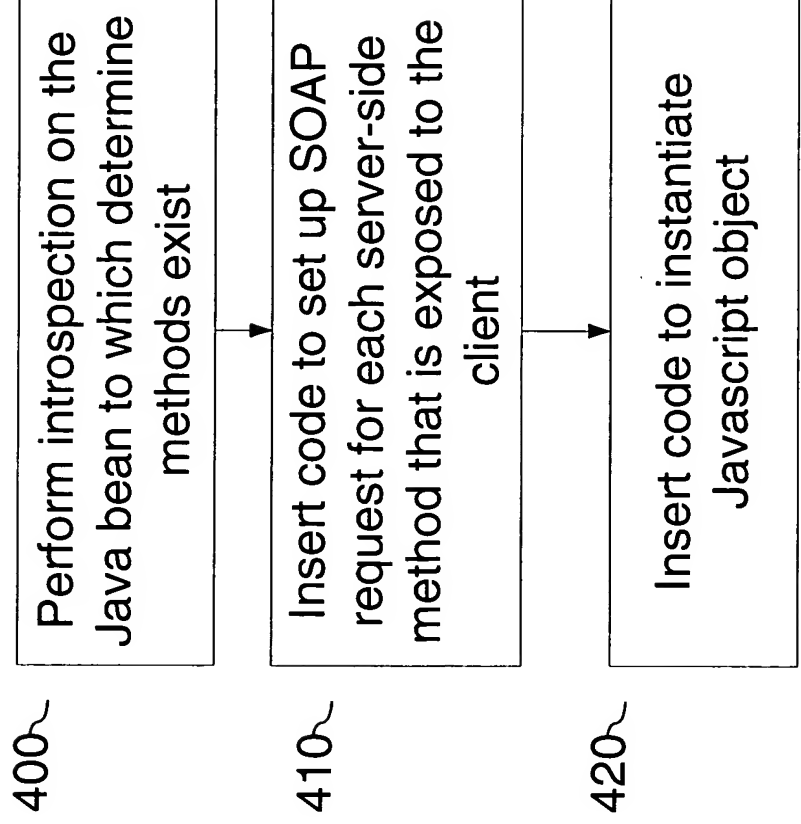


Fig. 4a

Automatic Generation of Javascript on Server-Side

430~

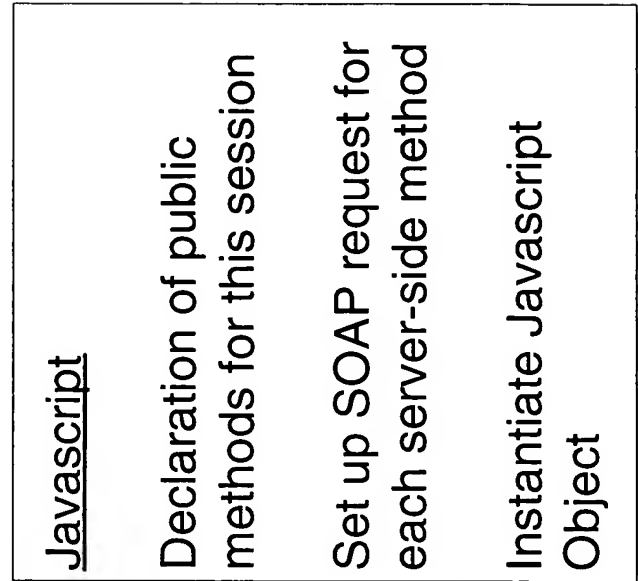


Fig. 4b

Functional Description of Automatically Generated Javascript

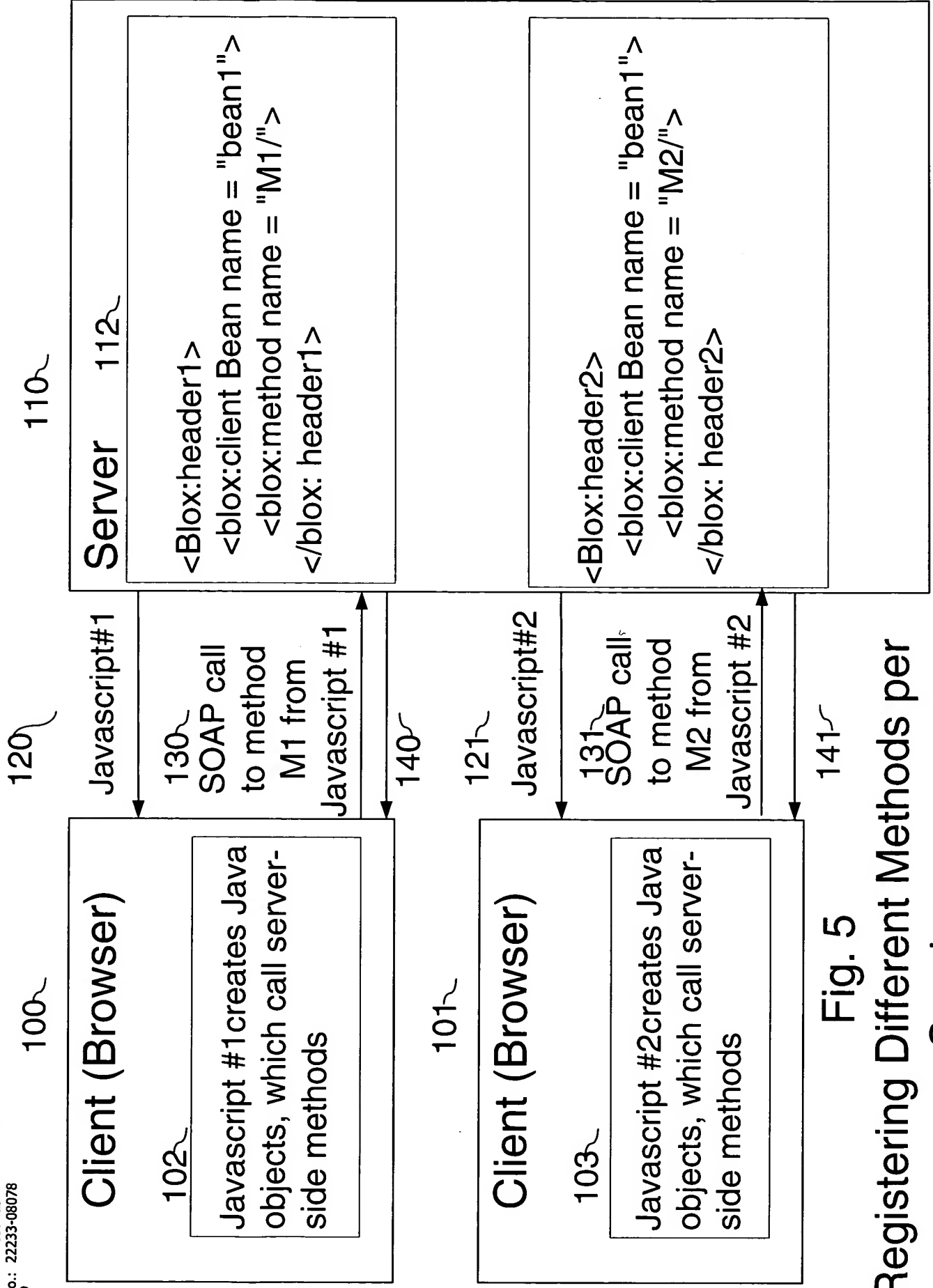


Fig. 5  
Registering Different Methods per  
Session

```

<%@ page import="com.alphablob.blox.PresentBlox,
com.alphablob.blox.ServerBloxException"%>
<%@ taglib uri='bloxtld' prefix='blox'%>
<%@ taglib uri='bloxutil' prefix='bloxui'%>
<blox:data id="calldata" dataSourceName="qcc-msas" useAliases="true" visible="false"
query="SELECT {(Time).[Calendar].[All Time Periods].[2000].[Qtr 3 00], [Scenario].[All Scenario].[All Time
Periods].[2000].[Qtr 4 00], [Scenario].[All Scenario].[Actual]], (Time).[Calendar].[All Time Periods].[2001], [Scenario].[All Scenario].[Budget]],
(Time).[Calendar].[All Time Periods].[2001].[Qtr 1 01], [Scenario].[All Scenario].[Budget]], (Time).[Calendar].[All Time Periods].[2001].[Qtr 2 01],
[Scenario].[All Scenario].[Budget]] ON COLUMNS, {[Locations].[All Locations], [Locations].[All Locations].Children} ON ROWS FROM QCC" />
<blox:present id="callpresent" width="600" height="500" menubarVisible="true" chartAvailable="false" visible="false">
<blox:toolbar visible="true" />
<blox:grid bandingEnabled="true" />
<blox:data bloxRef="calldata" />
</blox:present>
<%-- Create and register a J2EE session bean --%>
<jsp:useBean class="MyBean" scope="session" id="myBean">
<%
myBean.setBlox( callpresent );
%>
</jsp:useBean>
</html>
</head>
<%-- Register the bean as a client bean --%>
<blox:header>
<blox:clientBean name="myBean" />
</blox:header>
</script>
// Use ClientBean to invoke method on the bean
function showDataLayout( show )
{
// Call the server-side bean's method
var result = myBean.showDataLayout( show );

// Display the return value from the server-side bean method
alert( "Result type: " + typeof result + "\n\n" + result );
}
</script>
</head>
<body>
<input type="button" value="Hide Data Layout" onclick="showDataLayout( false );" >
<input type="button" value="Show Data Layout" onclick="showDataLayout( true );" >
</body>
</html>

```

600

602

604

606

608

Fig. 6(a)  
Example JSP Page

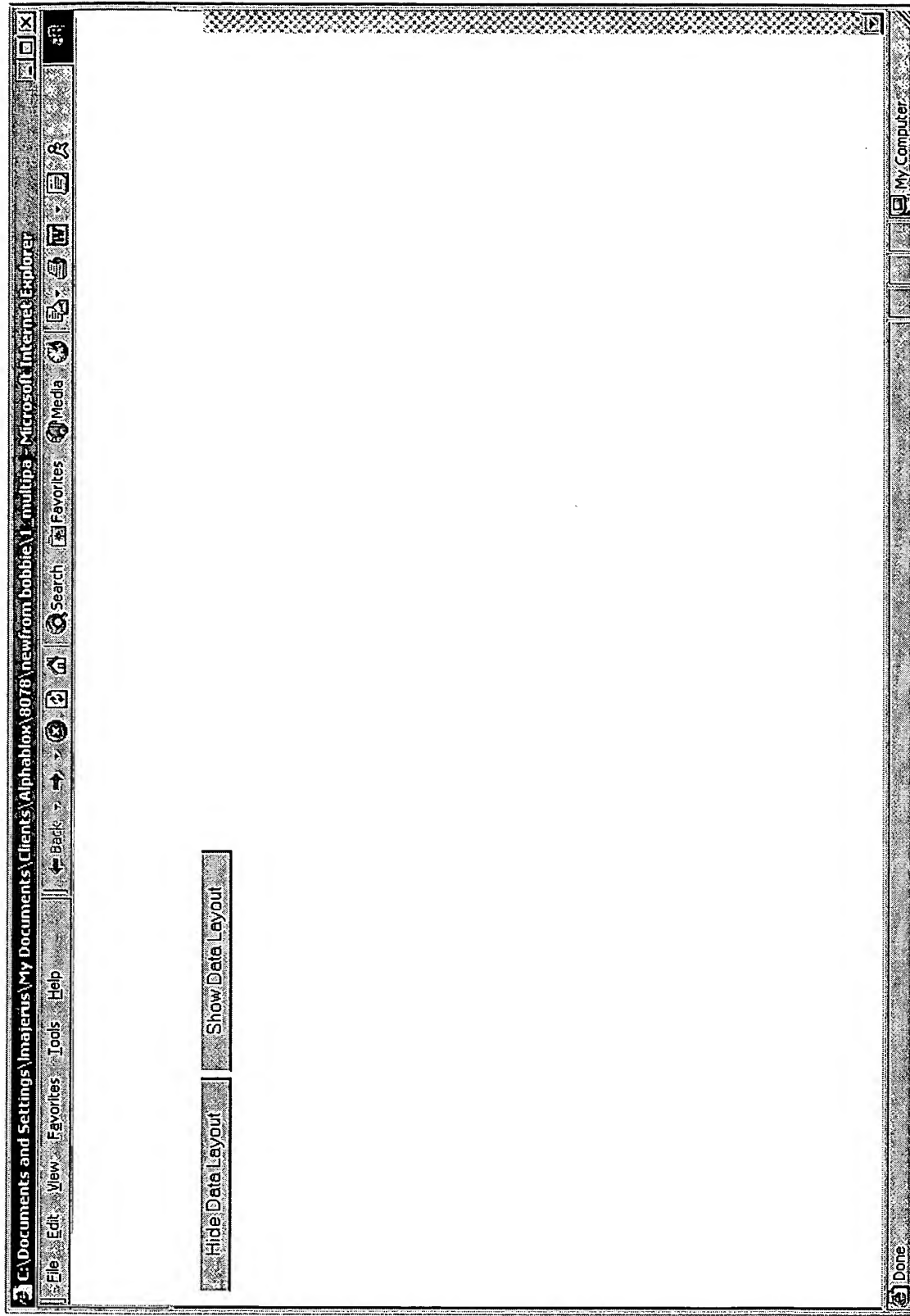


Fig. 6b

700

```

import com.alphablox.blox.PresentBlox;
import com.alphablox.blox.ServerBloxException;

MyBean.java

public class MyBean {
    private PresentBlox blox;

    public MyBean( )
    {
    }

    public void setBlox( final PresentBlox blox ) {
        this.blox = blox;
    }

    public String[] showDataLayout( boolean show ) {
        try {
            blox.setDataLayoutAvailable( show );
        } catch (ServerBloxException e) {
            e.printStackTrace();
        }

        return new String[] { "one", "two", "three" };
    }
}

```

Fig 7  
Example of bean

bean\_myBean.js  
/\* Automatically Generated JavaScript - do not modify \*/

function ImyBean()  
{

// Public methods

this.showDataLayout = \_showDataLayout;  
this.setBlox = \_setBlox;

function \_showDataLayout( arg1 ) {  
var soap = new SimpleSoapRequest( "callBean" );  
soap.addParameter( "bean", "myBean" );  
soap.addParameter( "method", "showDataLayout" );  
soap.addParameter( "arg1", arg1, "boolean" );  
return bloxAPI.rpc( soap );  
}

function \_setBlox( arg1 ) {  
var soap = new SimpleSoapRequest( "callBean" );  
soap.addParameter( "bean", "myBean" );  
soap.addParameter( "method", "setBlox" );  
soap.addParameter( "arg1", arg1 );  
return bloxAPI.rpc( soap );  
}

var myBean = new ImyBean( );  
document.myBean = myBean;

Fig 8

BEST AVAILABLE COPY



Express Mail Label No. EV 333134887 US  
Title: Automatically Creating Javascript  
Objects To Invoke Methods On Server-Side  
Java Beans  
Inventor(s): Robert DeSantis  
Atty. Docket No.: 22233-08078  
Sheet 9 of 10

<html>  
<head>  
  
<!-- START ALPHABLOX HEADER -->

900

<META HTTP-EQUIV="Pragma" CONTENT="no-cache">  
<LINK REL="stylesheet" HREF="/AlphabloxServer/theme/coleman/coleman.css" TYPE="text/css">  
<LINK REL="stylesheet" HREF="/AlphabloxServer/theme/coleman/coleman\_dhtml.css" TYPE="text/css">  
<LINK REL="stylesheet" HREF="/AlphabloxServer/theme/dhtml\_view.css" TYPE="text/css">

<SCRIPT>  
var \_appContext = "dca";  
var \_appName = "dca";  
var \_renderType = "dhtml";  
var \_pollInterval = 120000;  
var \_homeURL = "/dca/abx/peer/myBean.jsp?AAS\_QZ=1072744706967";  
var \_themeName = "coleman";  
var \_imageURL = "/AlphabloxServer/theme/coleman/i/";  
var \_prefixURL = "";  
var \_ibarArray = [];  
</SCRIPT>

<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/theme/aas.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/compatibility.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/stringUtilities.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/browserCheck.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/core.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/BloxHttp.js"></SCRIPT> 902  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/Soap.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/BloxAPI.js"></SCRIPT> 904  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/Events.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/Blox.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/Dialog.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/clientToolboxAPI.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/color.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/grid.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/cell.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/jslib/dhtml/utility.js"></SCRIPT>  
<SCRIPT LANGUAGE="javascript" src="/AlphabloxServer/temp/script/dca/bean\_myBean.js"></SCRIPT> 906

<!-- END ALPHABLOX HEADER -->

Fig. 9a  
HTML Sent to Server

<script>

// Use ClientBean to invoke method on the bean  
function showDataLayout( show )

{

// Call the server-side bean's method  
var result = myBean.showDataLayout( show );

// Display the return value from the server-side bean method  
alert( "Result type: " + typeof result + "\n\n" + result );

}

</script>

</head>

<body>

<input type="button" value="Hide Data Layout" onclick="showDataLayout( false );" >

<input type="button" value="Show Data Layout" onclick="showDataLayout( true );" >

</body>

</html>

Fig. 9b  
HTML Sent to Server  
(cont.)